Vulnerability Assessment and Penetration Testing Report of ESM

May 2023

L&T Technology Services

About L&T Technology Services:
L&T Technology Services Limited (LTTS) is a global leader in Engineering and R&D (ER&D) services. With 399 patents filed for 51 of the Global Top 100 ER&D spenders. Our innovations speak for itself – World's 1st Autonomous Welding Robot, Solar 'Connectivity' Drone, and the Smartest Campus in the World, to name a few. LTTS expertise in engineering design, product development, smart manufacturing, and digitalization touches every area of our lives. With 49 Innovation and R&D design centres globally, we specialize in disruptive technology spaces such as 5G, Artificial Intelligence, Collaborative Robots, Digital Factory, and Autonomous Transport.
LTTS is a publicly listed subsidiary of Larsen & Toubro Limited, the $18 billion Indian conglomerate operating in over 30 countries.

Document History:

Rel. No.	**Release Date**	**Prepared By. Prepared. Dt.**	**Reviewed By Reviewed Dt.**	**Approved By Approval Dt.**	**Remark/Revision Details**
1.0	May 2023	Sanidya Sharan Sai Praneetha Bhaskaruni	Sunil M.C		
		23/05/2023			

Table of Contents

1. Overview of the project

L&T Technology Services (LTTS) security team has conducted Security Assessment for the SmartCare Remote Management web application. The purpose of the assessment is to evaluate the security posture of the web application against common vulnerabilities.

Objective of the security assessment:
As a part of this engagement, a holistic approach was taken to conduct the Vulnerability Assessment and Penetration Testing on the ESM. During the engagement High, Medium, and Low severity issues were identified with respect to the ESM web application.

Approach
The following approach was taken to make sure the target was assessed against known vulnerabilities from all possible security perspectives:

- Manual Vulnerability Assessment and Penetration Testing using OWASP TOP 10 for web applications.

Some of the tools which were used are listed below:

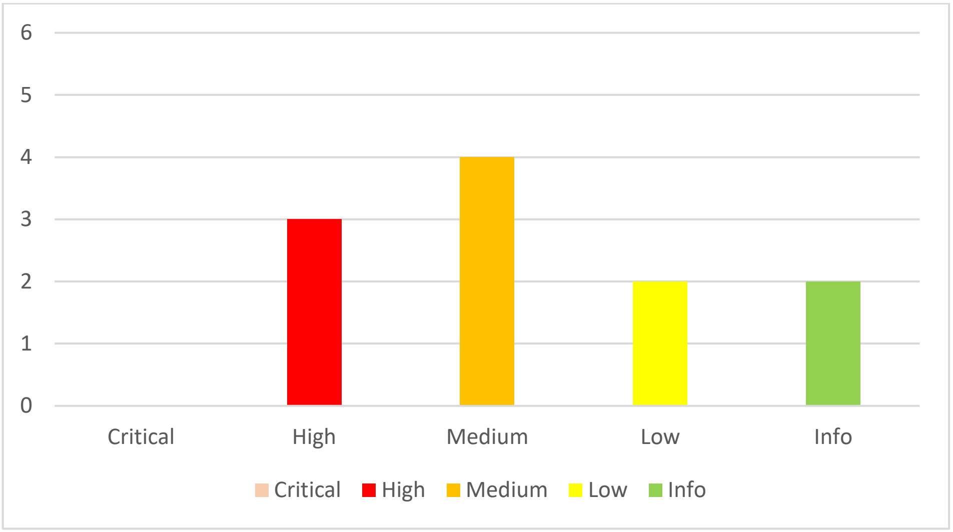
Target Application	ESM
Browser	Chrome, Firefox
Tools	Burp Suite, Postman, Wireshark

Key Security Policies
OWASP top 10 listed vulnerabilities were used as a reference framework. The following key security aspects were checked:

1. Broken Access Control
2. Cryptographic Failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery

Summary of Findings

The graph below shows a summary of the number of vulnerabilities found for each impact level for the Application Security Assessment. Vulnerabilities found are addressed according to priority, findings, analysis, and recommendations from the assessment.



Sr.no	Title	Risk Rating
1	Application Accessible over HTTP	High
2	No Account Lockout Policy	High
3	Cleartext Transmission of Sensitive Information	High
4	Blind Cross-site Scripting (XSS)	Medium
5	Clickjacking	Medium
6	HTML Injection	Medium
7	Lack of rate limit leads to Brute-force Attack	Medium
8	No Content Security Policy	Low
9	Application Allows Concurrent Sessions	Low
10	No Multifactor Authentication (MFA)	Info
11	Long Session Timeout	Info

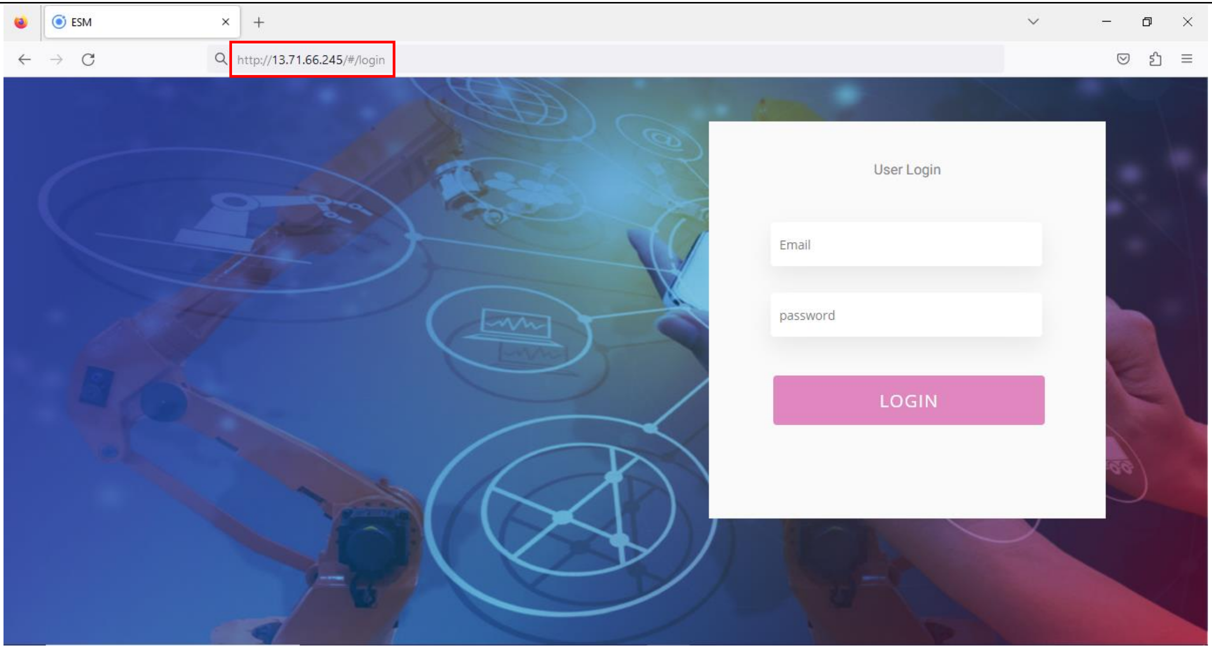
Overall Risk Severity = Likelihood x Impact

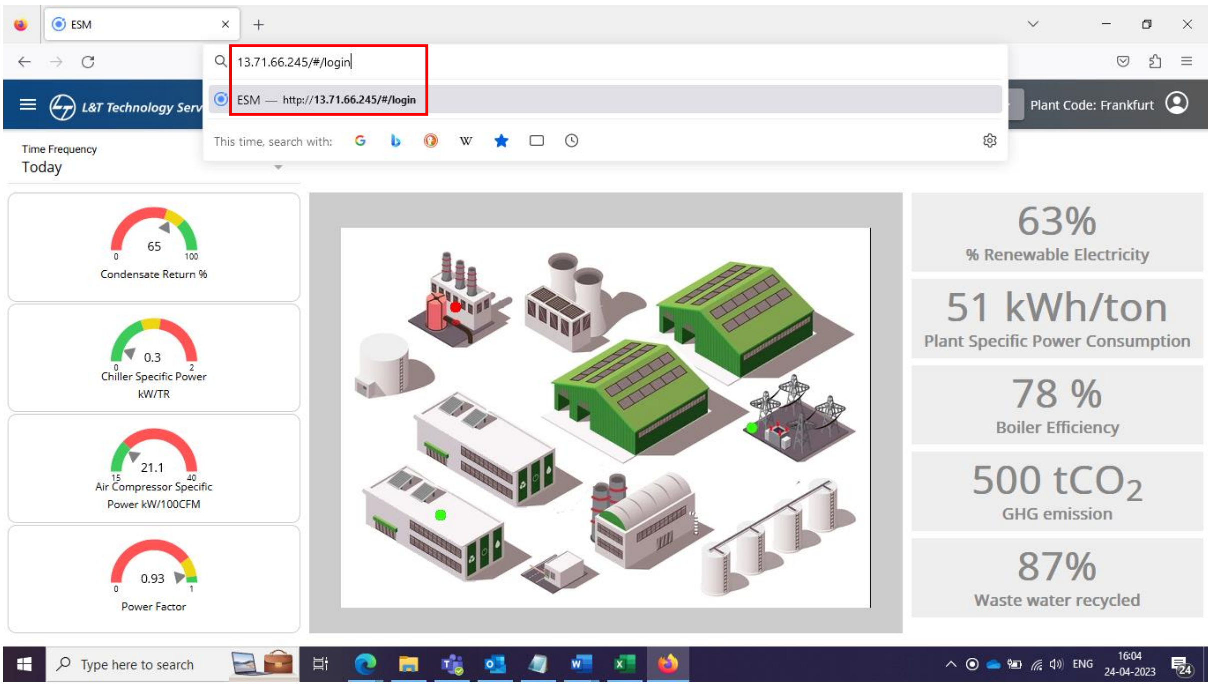
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

L&T Technology Services

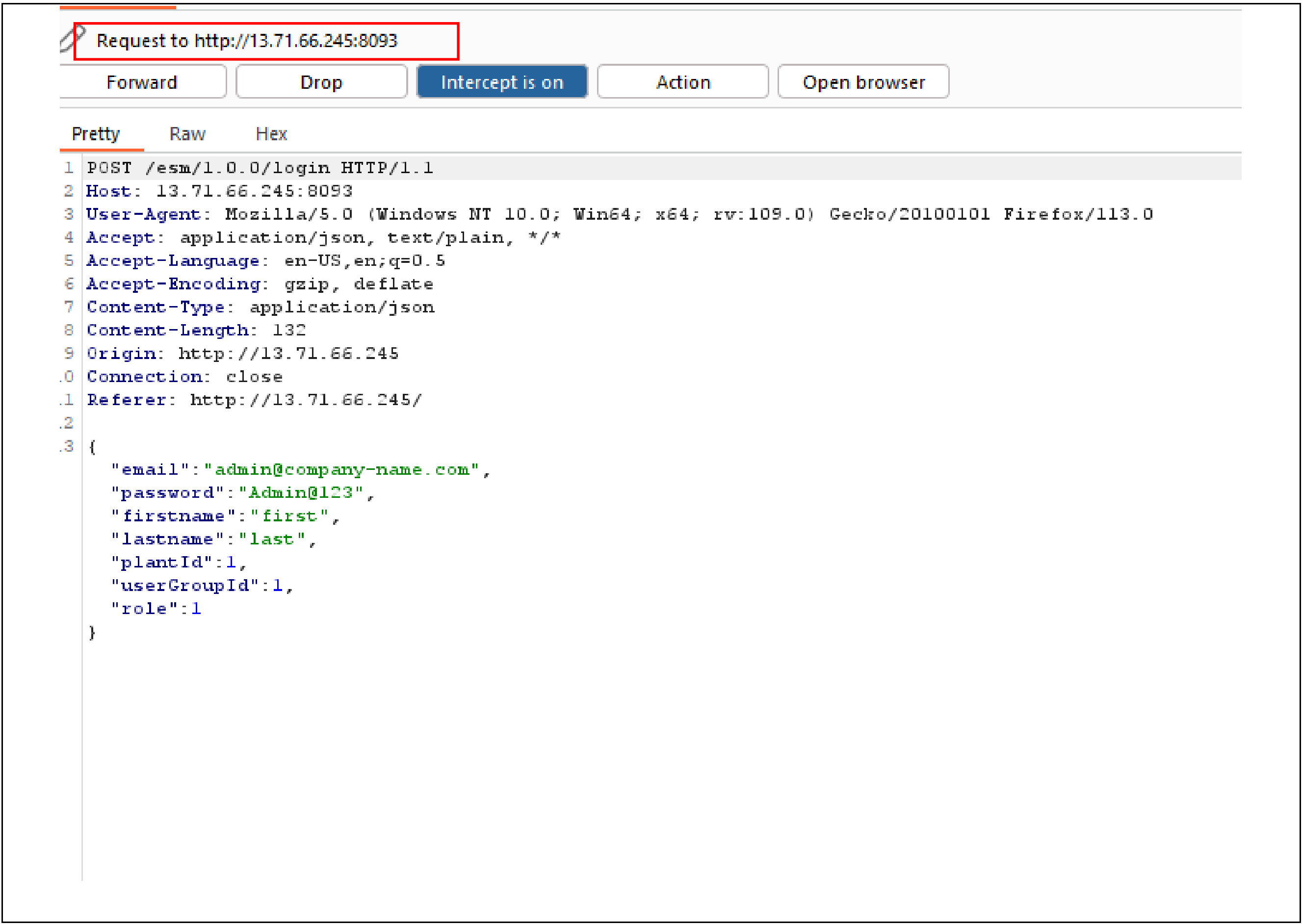
2. Vulnerabilities explained in detail

2.1 Application Accessible over HTTP			
Impact	High	**Risk Rating**	High
Ease of Exploit	Easy		
Likelihood	Medium		
Category	CWE-319: Cleartext Transmission of Sensitive Information		
URL/Impacted system	http://13.71.66.245/#/login		
Description			
During vulnerability assessment, ESM web application allows web browsers to access to the application over HTTP and doesn't redirect them to HTTPS. The application fails to prevent users from connecting to it over unencrypted connections.			
Impact			
• This is performed by rewriting HTTPS links as HTTP, so that if a targeted user follows a link to the site from an HTTP page, their browser never attempts to use an encrypted connection. • An attacker able to modify a legitimate user's network traffic could bypass the application's use of SSL/TLS encryption and use the application as a platform for attacks against its users.			
Recommendation			
• The application should instruct web browsers to only access the application using HTTPS. • Enable HTTP Strict Transport Security (HSTS) by adding a response header with the name 'Strict-Transport-Security' and the value 'max-age=expireTime', where expireTime is the time in seconds that browsers should remember that the site should only be accessed.			
How to recreate the Security defect			
• Browse to – http://13.71.66.245/#/login • Application does not redirect to HTTPS and the traffic is sent over HTTP.			
Evidence			





L&T Technology Services



```
POST /esm/1.0.0/login HTTP/1.1
Host: 13.71.66.245:8093
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/113.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 132
Origin: http://13.71.66.245
Connection: close
Referer: http://13.71.66.245/

{
  "email":"admin@company-name.com",
  "password":"Admin@123",
  "firstname":"first",
  "lastname":"last",
  "plantId":1,
  "userGroupId":1,
  "role":1
}
```

2.2 No Account Lockout Policy

Impact	High	**Risk Rating**	🟥 High
Ease of Exploit	Moderate		
Likelihood	Medium		
Category	Broken Authentication		
URL/Impacted system	http://13.71.66.245/#/login		

Description

During Assessment, we observed ESM web application doesn't follow the account lockout policy. Current mechanism allows for N number of attempts to occur on any username. This gives attackers the bandwidth needed to carry out brute force attacks. If the attackers know the username, then the brute force on password field can happen with increased probability. If the attackers do not know the username, then they can still configure utilities to run dictionary based or other variants to run on ESM for longer durations.

Impact

It is possible for an attacker to gain access to the application by brute-forcing the password. Since there are no restrictions on the number of logins attempts a malicious user can brute force the credentials of a user until the right credentials are guessed.

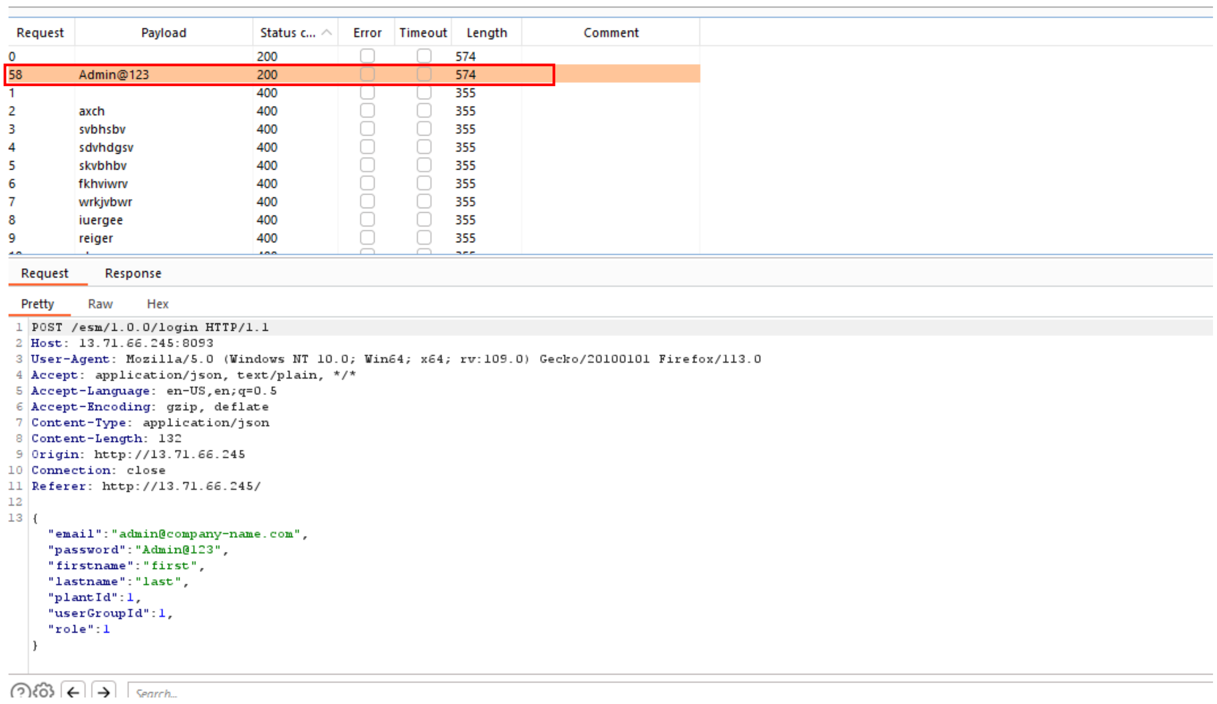
Recommendation

The most obvious way to block brute-force attacks is to simply lock out accounts after a defined number of incorrect password attempts. Account lockouts can last a specific duration, such as one hour, or the accounts could remain locked until manually unlocked by an administrator. A CAPTCHA may hinder brute force attacks, but CAPTCHA should be perceived as a rate limiting protection only which stops the attacker for a limited amount of time, also can use alternative verification channels like multi factor authentication.

How to recreate the Security defect

• Login into the ESM web application.
• Configure Burp or any other utility to continuously execute login payload with input coming from a dictionary file.
• We observed that login into the account without any lockout.

Evidence



Request	Payload	Status c...	Error	Timeout	Length	Comment
0		200	☐	☐	574	
58	Admin@123	200	☐	☐	574	
1		400	☐	☐	355	
2	axch	400	☐	☐	355	
3	svbhsbv	400	☐	☐	355	
4	sdvhdgsv	400	☐	☐	355	
5	skvbhbv	400	☐	☐	355	
6	fkhviwrv	400	☐	☐	355	
7	wrkjvbwr	400	☐	☐	355	
8	iuergee	400	☐	☐	355	
9	reiger	400	☐	☐	355	

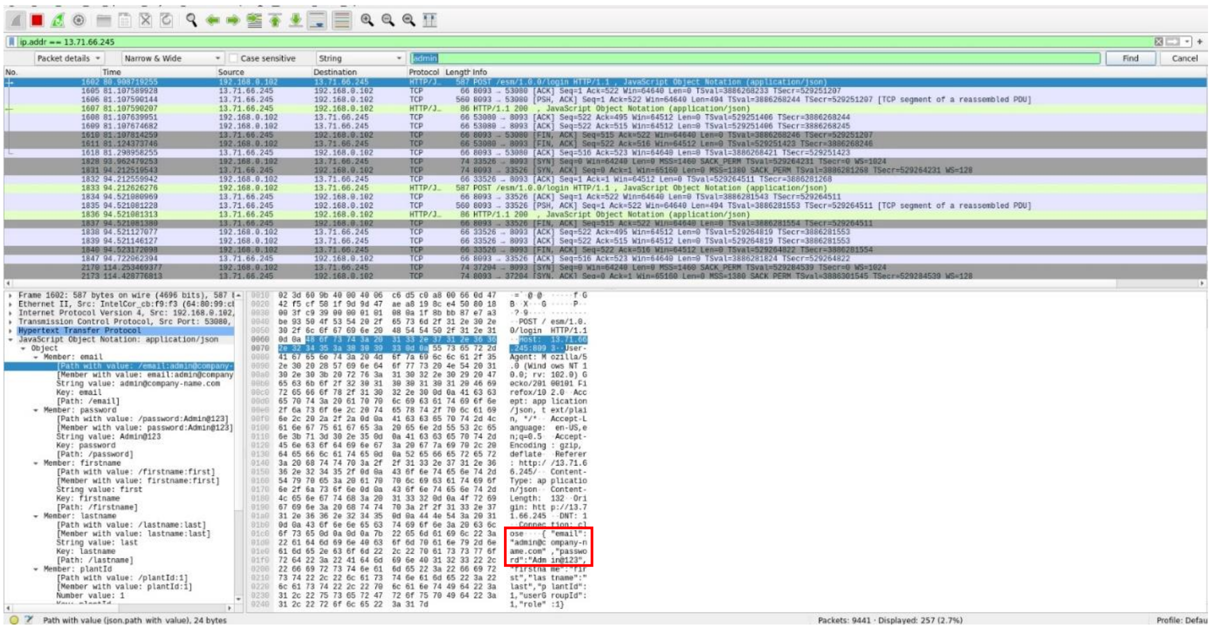
```
POST /esm/1.0.0/login HTTP/1.1
Host: 13.71.66.245:8093
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/113.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 132
Origin: http://13.71.66.245
Connection: close
Referer: http://13.71.66.245/

{
  "email":"admin@company-name.com",
  "password":"Admin@123",
  "firstname":"first",
  "lastname":"last",
  "plantId":1,
  "userGroupId":1,
  "role":1
}
```

2.3 Cleartext Transmission of Sensitive Information

Impact	Medium	**Risk Rating**	High
Ease of Exploit	Easy		
Likelihood	High		
Category	CWE-319: Cleartext Transmission of Sensitive Information		
URL/Impacted system	http://13.71.66.245/#/login		

Description

Cleartext transmission, also known as plaintext transmission, refers to the process of transmitting data over a network or communication channel without encryption or other security measures that protect the data from interception or unauthorized access.

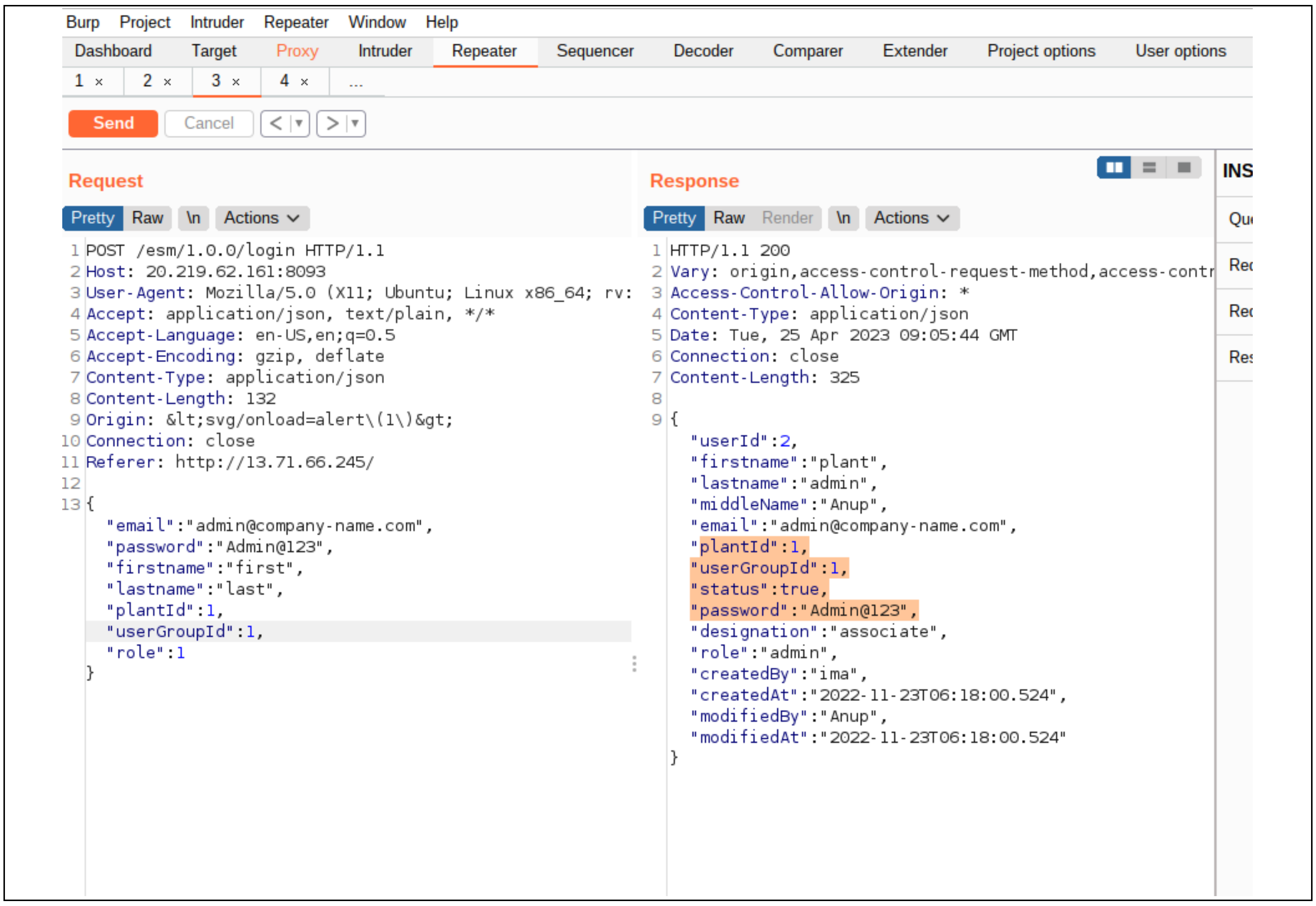
Impact
• The attacker can steal the user credentials and it can lead to identity theft. • Information disclosure is possible. • Data tampering is possible.
Recommendation
• Use encryption protocols such as SSL/TLS or HTTPS to ensure that data is encrypted in transit. • Use secure communication channels such as SFTP, SSH, or VPNs to transmit sensitive data. • Disable cleartext protocols such as HTTP or FTP and use only encrypted protocols such as HTTPS or SFTP to transmit sensitive data.
How to recreate the Security defect
• Open Wireshark • Enter the IP address as ip.addr == 13.71.66.245 and select the internet connectivity interface. • Open the packets and you can see sensitive information in clear text.
Evidence


```
ip.addr == 13.71.66.245
Packet details   Narrow & Wide   Case sensitive   String   admin   Find   Cancel

No.   Time           Source         Destination    Protocol Length Info
1602  80.908719255   192.168.0.102  13.71.66.245   HTTP/J…  587 POST /esm/1.0.0/login HTTP/1.1 , JavaScript Object Notation (application/json)
1605  81.107589928   13.71.66.245   192.168.0.102  TCP       66 8093 → 53080 [ACK] Seq=1 Ack=522 Win=64640 Len=0 TSval=3886268233 TSecr=529251207
1606  81.107590144   13.71.66.245   192.168.0.102  TCP      560 8093 → 53080 [PSH, ACK] Seq=1 Ack=522 Win=64640 Len=494 TSval=3886268244 TSecr=529251207 [TCP segment of a reassembled PDU]
1607  81.107590207   13.71.66.245   192.168.0.102  HTTP/J…   86 HTTP/1.1 200  , JavaScript Object Notation (application/json)
1608  81.107639951   192.168.0.102  13.71.66.245   TCP       66 53080 → 8093 [ACK] Seq=522 Ack=495 Win=64512 Len=0 TSval=529251406 TSecr=3886268244
1609  81.107674682   192.168.0.102  13.71.66.245   TCP       66 53080 → 8093 [ACK] Seq=522 Ack=515 Win=64512 Len=0 TSval=529251406 TSecr=3886268245
1610  81.107814259   13.71.66.245   192.168.0.102  TCP       66 8093 → 53080 [FIN, ACK] Seq=515 Ack=522 Win=64640 Len=0 TSval=3886268246 TSecr=529251207
1611  81.124373746   192.168.0.102  13.71.66.245   TCP       66 53080 → 8093 [FIN, ACK] Seq=522 Ack=516 Win=64512 Len=0 TSval=529251423 TSecr=3886268246
1618  81.298958255   13.71.66.245   192.168.0.102  TCP       66 8093 → 53080 [ACK] Seq=516 Ack=523 Win=64640 Len=0 TSval=3886268421 TSecr=529251423
1828  93.962479253   192.168.0.102  13.71.66.245   TCP       74 33526 → 8093 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=529264231 TSecr=0 WS=1024
1831  94.212519543   13.71.66.245   192.168.0.102  TCP       74 8093 → 33526 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1380 SACK_PERM TSval=3886281268 TSecr=529264231 WS=128
1832  94.212559942   192.168.0.102  13.71.66.245   TCP       66 33526 → 8093 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=529264511 TSecr=3886281268
1833  94.212626276   192.168.0.102  13.71.66.245   HTTP/J…  587 POST /esm/1.0.0/login HTTP/1.1 , JavaScript Object Notation (application/json)
1834  94.521080969   13.71.66.245   192.168.0.102  TCP       66 8093 → 33526 [ACK] Seq=1 Ack=522 Win=64640 Len=0 TSval=3886281543 TSecr=529264511
1835  94.521081228   13.71.66.245   192.168.0.102  TCP      560 8093 → 33526 [PSH, ACK] Seq=1 Ack=522 Win=64640 Len=494 TSval=3886281553 TSecr=529264511 [TCP segment of a reassembled PDU]
1836  94.521081313   13.71.66.245   192.168.0.102  HTTP/J…   86 HTTP/1.1 200  , JavaScript Object Notation (application/json)
1837  94.521081380   13.71.66.245   192.168.0.102  TCP       66 8093 → 33526 [FIN, ACK] Seq=515 Ack=522 Win=64640 Len=0 TSval=3886281554 TSecr=529264511
1838  94.521127077   192.168.0.102  13.71.66.245   TCP       66 33526 → 8093 [ACK] Seq=522 Ack=495 Win=64512 Len=0 TSval=529264819 TSecr=3886281553
1839  94.521146127   192.168.0.102  13.71.66.245   TCP       66 33526 → 8093 [ACK] Seq=522 Ack=515 Win=64512 Len=0 TSval=529264819 TSecr=3886281553
1840  94.523172098   192.168.0.102  13.71.66.245   TCP       66 33526 → 8093 [FIN, ACK] Seq=522 Ack=516 Win=64512 Len=0 TSval=529264822 TSecr=3886281554
1847  94.722062394   13.71.66.245   192.168.0.102  TCP       66 8093 → 33526 [ACK] Seq=516 Ack=523 Win=64640 Len=0 TSval=3886281824 TSecr=529264822
2170 114.253469377   192.168.0.102  13.71.66.245   TCP       74 37204 → 8093 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=529284539 TSecr=0 WS=1024
2173 114.428776813   13.71.66.245   192.168.0.102  TCP       74 8093 → 37204 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1380 SACK_PERM TSval=3886301545 TSecr=529284539 WS=128
```

```
▸ Frame 1602: 587 bytes on wire (4696 bits), 587 b
▸ Ethernet II, Src: IntelCor_cb:f9:f3 (64:80:99:cb
▸ Internet Protocol Version 4, Src: 192.168.0.102,
▸ Transmission Control Protocol, Src Port: 53080,
▸ Hypertext Transfer Protocol
▾ JavaScript Object Notation: application/json
  ▾ Object
    ▾ Member: email
        [Path with value: /email:admin@company-
        [Member with value: email:admin@company
        String value: admin@company-name.com
        Key: email
        [Path: /email]
    ▾ Member: password
        [Path with value: /password:Admin@123]
        [Member with value: password:Admin@123]
        String value: Admin@123
        Key: password
        [Path: /password]
    ▾ Member: firstname
        [Path with value: /firstname:first]
        [Member with value: firstname:first]
        String value: first
        Key: firstname
        [Path: /firstname]
    ▾ Member: lastname
        [Path with value: /lastname:last]
        [Member with value: lastname:last]
        String value: last
        Key: lastname
        [Path: /lastname]
    ▾ Member: plantId
        [Path with value: /plantId:1]
        [Member with value: plantId:1]
        Number value: 1
```

```
0010  02 3d 60 9b 40 00 40 06  c6 d5 c0 a8 00 66 0d 47   ·=`·@·@· ·····f·G
0020  42 f5 cf 58 1f 9d 9d 47  ae a8 19 8c e4 50 80 18   B··X···G ·····P··
0030  00 3f c9 39 00 00 01 01  08 0a 1f 8b bb 87 e7 a3   ·?·9···· ········
0040  be 93 50 4f 53 54 20 2f  65 73 6d 2f 31 2e 30 2e   ··POST / esm/1.0.
0050  30 2f 6c 6f 67 69 6e 20  48 54 54 50 2f 31 2e 31   0/login  HTTP/1.1
0060  0d 0a 48 6f 73 74 3a 20  31 33 2e 37 31 2e 36 36   ··Host:  13.71.66
0070  2e 32 34 35 3a 38 30 39  33 0d 0a 55 73 65 72 2d   .245:809 3··User-
0080  41 67 65 6e 74 3a 20 4d  6f 7a 69 6c 6c 61 2f 35   Agent: M ozilla/5
0090  2e 30 20 28 57 69 6e 64  6f 77 73 20 4e 54 20 31   .0 (Wind ows NT 1
00a0  30 2e 30 3b 20 72 76 3a  31 30 32 2e 30 29 20 47   0.0; rv: 102.0) G
00b0  65 63 6b 6f 2f 32 30 31  30 30 31 30 31 20 46 69   ecko/201 00101 Fi
00c0  72 65 66 6f 78 2f 31 30  32 2e 30 0d 0a 41 63 63   refox/10 2.0··Acc
00d0  65 70 74 3a 20 61 70 70  6c 69 63 61 74 69 6f 6e   ept: app lication
00e0  2f 6a 73 6f 6e 2c 20 74  65 78 74 2f 70 6c 61 69   /json, t ext/plai
00f0  6e 2c 20 2a 2f 2a 0d 0a  41 63 63 65 70 74 2d 4c   n, */*·· Accept-L
0100  61 6e 67 75 61 67 65 3a  20 65 6e 2d 55 53 2c 65   anguage:  en-US,e
0110  6e 3b 71 3d 30 2e 35 0d  0a 41 63 63 65 70 74 2d   n;q=0.5· ·Accept-
0120  45 6e 63 6f 64 69 6e 67  3a 20 67 7a 69 70 2c 20   Encoding : gzip,
0130  64 65 66 6c 61 74 65 0d  0a 52 65 66 65 72 65 72   deflate· ·Referer
0140  3a 20 68 74 74 70 3a 2f  2f 31 33 2e 37 31 2e 36   : http:/ /13.71.6
0150  36 2e 32 34 35 2f 0d 0a  43 6f 6e 74 65 6e 74 2d   6.245/·· Content-
0160  54 79 70 65 3a 20 61 70  70 6c 69 63 61 74 69 6f   Type: ap plicatio
0170  6e 2f 6a 73 6f 6e 0d 0a  43 6f 6e 74 65 6e 74 2d   n/json·· Content-
0180  4c 65 6e 67 74 68 3a 20  31 33 32 0d 0a 4f 72 69   Length:  132··Ori
0190  67 69 6e 3a 20 68 74 74  70 3a 2f 2f 31 33 2e 37   gin: htt p://13.7
01a0  31 2e 36 36 2e 32 34 35  0d 0a 44 4e 54 3a 20 31   1.66.245 ··DNT: 1
01b0  0d 0a 43 6f 6e 6e 65 63  74 69 6f 6e 3a 20 63 6c   ··Connec tion: cl
01c0  6f 73 65 0d 0a 0d 0a 7b  22 65 6d 61 69 6c 22 3a   ose····{ "email":
01d0  22 61 64 6d 69 6e 40 63  6f 6d 70 61 6e 79 2d 6e   "admin@c ompany-n
01e0  61 6d 65 2e 63 6f 6d 22  2c 22 70 61 73 73 77 6f   ame.com" ,"passwo
01f0  72 64 22 3a 22 41 64 6d  69 6e 40 31 32 33 22 2c   rd":"Adm in@123",
0200  22 66 69 72 73 74 6e 61  6d 65 22 3a 22 66 69 72   "firstna me":"fir
0210  73 74 22 2c 22 6c 61 73  74 6e 61 6d 65 22 3a 22   st","las tname":"
0220  6c 61 73 74 22 2c 22 70  6c 61 6e 74 49 64 22 3a   last","p lantId":
0230  31 2c 22 75 73 65 72 47  72 6f 75 70 49 64 22 3a   1,"userG roupId":
0240  31 2c 22 72 6f 6c 65 22  3a 31 7d                  1,"role" :1}
```

Path with value (json.path_with_value), 24 bytes   Packets: 9441 · Displayed: 257 (2.7%)   Profile: Defau


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Send Cancel
Request
Pretty Raw \n Actions
Response
Pretty Raw Render \n Actions
INS

```
POST /esm/1.0.0/login HTTP/1.1
Host: 20.219.62.161:8093
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 132
Origin: &lt;svg/onload=alert\(1\)&gt;
Connection: close
Referer: http://13.71.66.245/

{
  "email":"admin@company-name.com",
  "password":"Admin@123",
  "firstname":"first",
  "lastname":"last",
  "plantId":1,
  "userGroupId":1,
  "role":1
}
```

```
HTTP/1.1 200
Vary: origin,access-control-request-method,access-contr
Access-Control-Allow-Origin: *
Content-Type: application/json
Date: Tue, 25 Apr 2023 09:05:44 GMT
Connection: close
Content-Length: 325

{
  "userId":2,
  "firstname":"plant",
  "lastname":"admin",
  "middleName":"Anup",
  "email":"admin@company-name.com",
  "plantId":1,
  "userGroupId":1,
  "status":true,
  "password":"Admin@123",
  "designation":"associate",
  "role":"admin",
  "createdBy":"ima",
  "createdAt":"2022-11-23T06:18:00.524",
  "modifiedBy":"Anup",
  "modifiedAt":"2022-11-23T06:18:00.524"
}
```

2.4 Blind Cross-site Scripting (XSS)

Impact	Medium	**Risk Rating**	Medium
Ease of Exploit	Easy		
Likelihood	Medium		
Category	CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')		
URL/Impacted system	http://13.71.66.245/#/login		

Description

Blind XSS vulnerabilities are a variant of persistent XSS vulnerabilities. They occur when the attacker input is saved by the web server and executed as a malicious script in another part of the application or in another application. During penetration testing, we have generated a Blind XSS Alert. The JavaScript code that had been injected into the Alert name has been executed successfully. We can see the machines visiting on the alert page and clicking on the malicious alert name as shown in the evidence.

Impact

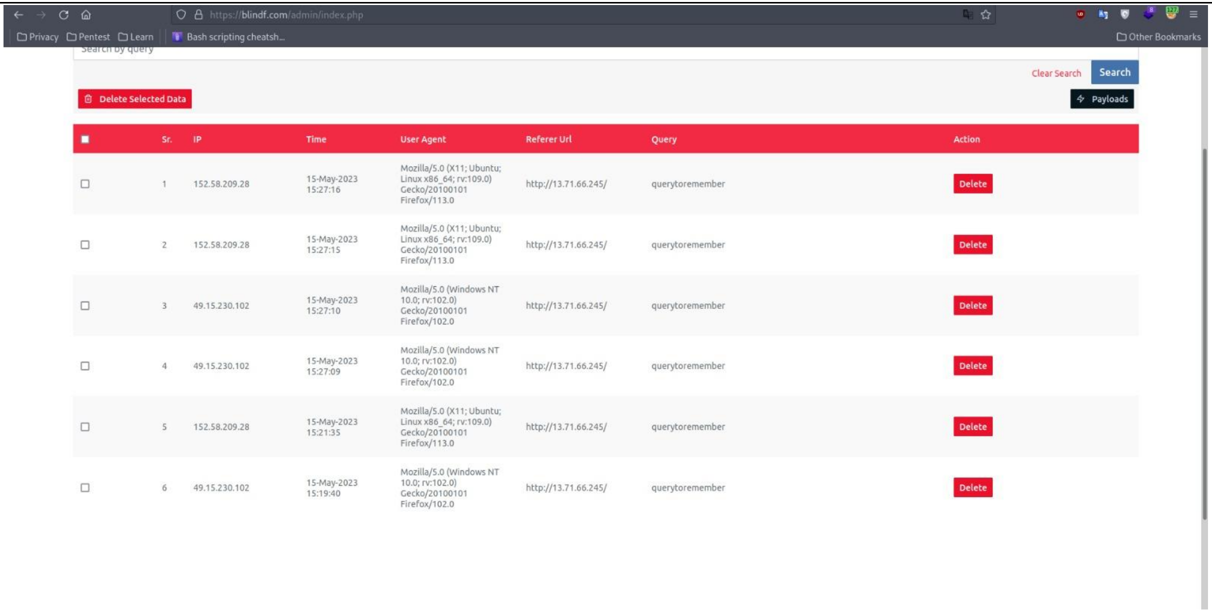
There are many different attacks that can be leveraged through the use of cross-site scripting, including:

• Attacking the user/machine which is clicking on the Alert name.
• Mounting phishing attacks

Recommendation

L&T Technology Services

• In Alert manager page, put filters for saving the Alert name. • Validate to catch potentially malicious user-provided input. • Use Content Security Policy (CSP) to reduce the severity of any XSS vulnerabilities that still occur.
How to recreate the Security defect
• Browse to – http://13.71.66.245/#/login • Go to alert manager page and create new alert. • In the alert name put a malicious XSS script and save it. • Then you will get a clickable alert name.
Evidence

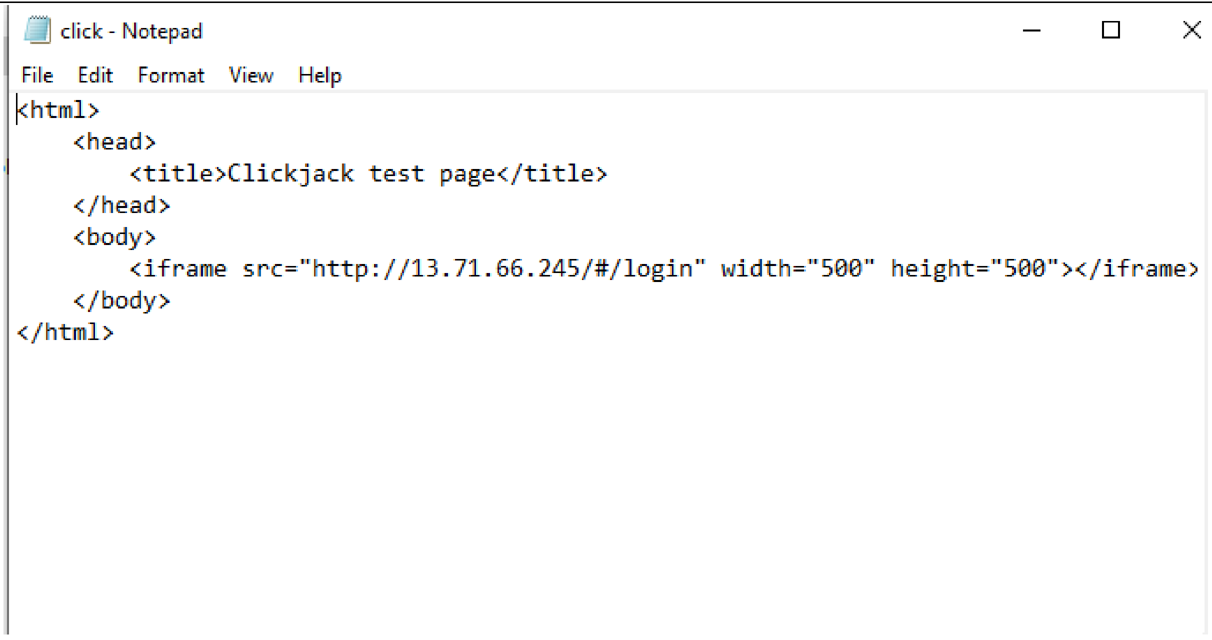


	Sr.	IP	Time	User Agent	Referer Url	Query	Action
☐	1	152.58.209.28	15-May-2023 15:27:16	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/113.0	http://13.71.66.245/	querytoremember	Delete
☐	2	152.58.209.28	15-May-2023 15:27:15	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/113.0	http://13.71.66.245/	querytoremember	Delete
☐	3	49.15.230.102	15-May-2023 15:27:10	Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0	http://13.71.66.245/	querytoremember	Delete
☐	4	49.15.230.102	15-May-2023 15:27:09	Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0	http://13.71.66.245/	querytoremember	Delete
☐	5	152.58.209.28	15-May-2023 15:21:35	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/113.0	http://13.71.66.245/	querytoremember	Delete
☐	6	49.15.230.102	15-May-2023 15:19:40	Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0	http://13.71.66.245/	querytoremember	Delete

2.5 Clickjacking			
Impact	Medium	**Risk Rating**	Medium
Ease of Exploit	Easy		
Likelihood	Medium		
Category	A6: Security Misconfiguration		
URL/Impacted system	http://13.71.66.245/#/login		
Description			
Clickjacking is a malicious technique that consists of deceiving a web user into interacting by clicking with something different to what the user believes they are interacting with. This type of attack, that can be used alone or in combination with other attacks, could potentially send unauthorized commands or reveal confidential information while the victim is interacting with seemingly harmless web pages.			
Impact			

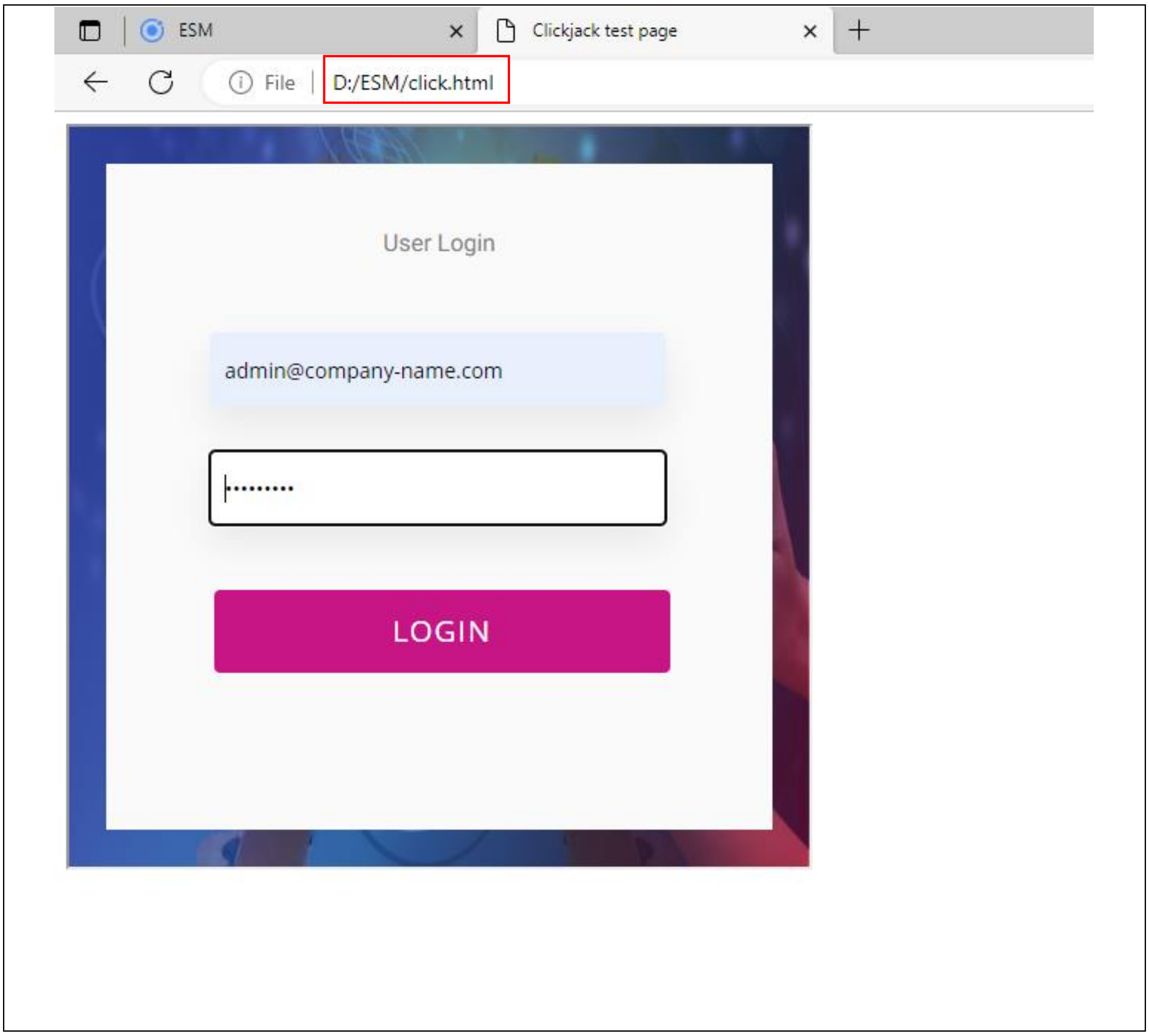
L&T Technology Services

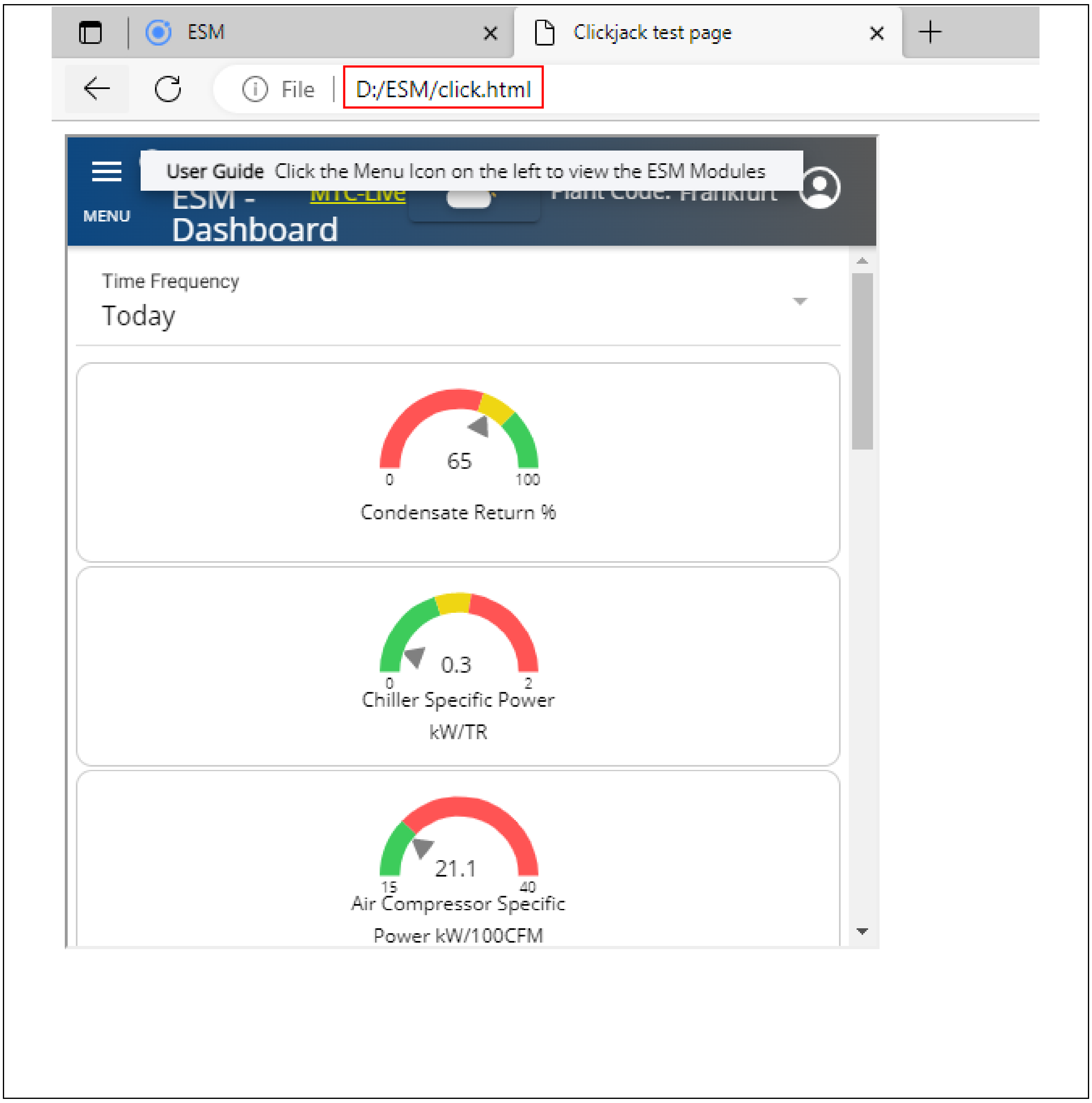
Attacker loads frame with high opacity onto the victim user's application page, something which is not the same what the user believed to be interacting with. Proof of clickjacking instance recorded is attached in the Evidence.
Recommendation
• The X-Frame-Options HTTP response header can be used to indicate whether a browser should be allowed to render a page in a <frame>, <iframe> or <object>. • Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites.
How to recreate the Security defect
• Write the following code into a notepad and save it as clickjacking.html. • Open that in the browser.
Evidence



```
<html>
    <head>
        <title>Clickjack test page</title>
    </head>
    <body>
        <iframe src="http://13.71.66.245/#/login" width="500" height="500"></iframe>
    </body>
</html>
```

L&T Technology Services

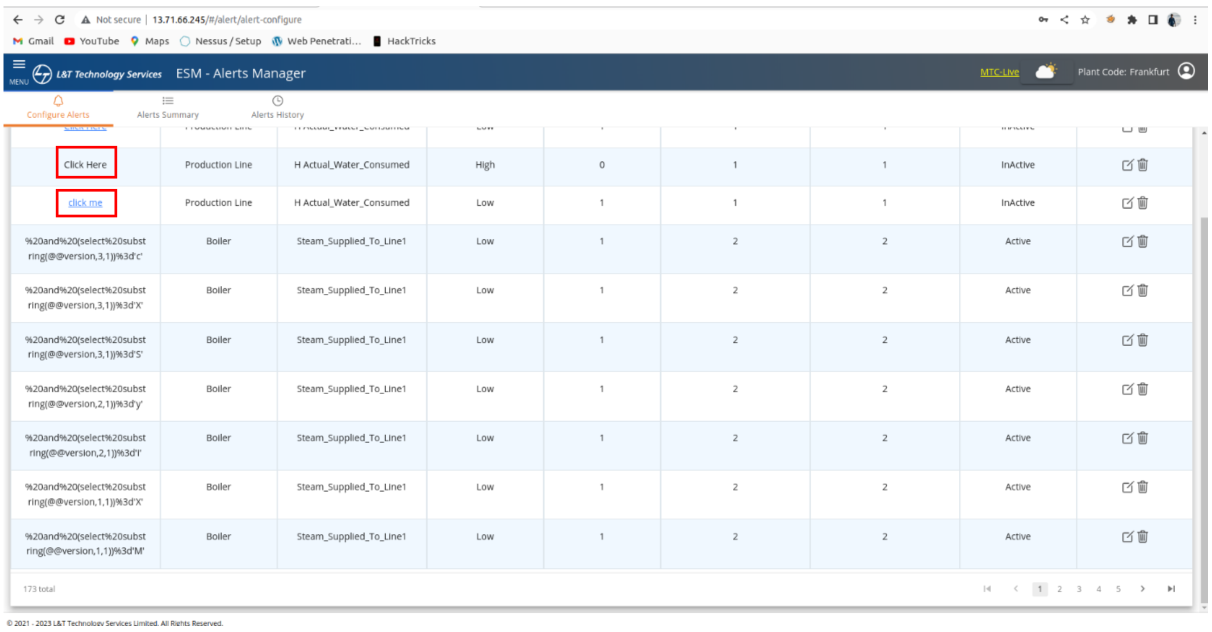

ESM
Clickjack test page
File
D:/ESM/click.html
User Login
admin@company-name.com
LOGIN



2.6 HTML Injection			
Impact	Medium	**Risk Rating**	Medium
Ease of Exploit	Moderate		
Likelihood	Medium		
Category	CWE-80: Improper Neutralization of script-Related HTML Tags in a Web Page		
URL/Impacted system	http://13.71.66.245/#/login		
Description			

L&T Technology Services

HTML injection is a type of injection vulnerability that occurs when a user is able to control an input point and is able to inject arbitrary HTML code into a vulnerable web page. Here, in alert manager tab, we got that entry point on the alert name option.
Impact
• It can allow an attacker to modify the page. • To steal another person’s identity. • The attacker discovers injection vulnerability and decides to use an HTML injection attack. • Attacker crafts malicious links, including his injected HTML content, and sends it to a user via email. • The user visits the page due to the page being located within a trusted domain. • The attacker’s injected HTML is rendered and presented to the user asking for a username and password. • The user enters a username and password, which are both sent to the attacker’s server.
Recommendation
• In Alert manager page, put filters for saving the Alert name. • Set up HTML script which filters the metacharacters from user inputs. • Implement functions to validate the user inputs such that they do not contain any specific tag that can lead to virtual defacements.
How to recreate the Security defect
• Open alert manager page. • Save new alert with one HTML script in the alert name box. • After saving, click on the alert name.
Evidence

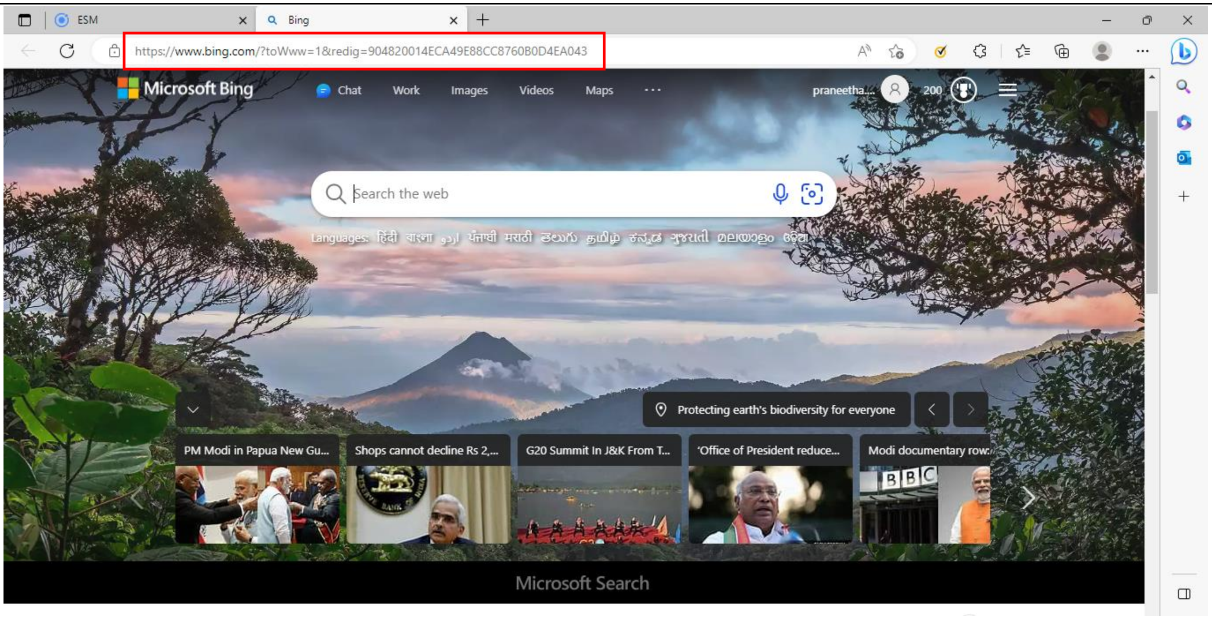


Click Here	Production Line	H Actual_Water_Consumed	High	0	1	1	InActive	
click me	Production Line	H Actual_Water_Consumed	Low	1	1	1	InActive	
%20and%20(select%20substring(@@version,3,1))%3d'c'	Boiler	Steam_Supplied_To_Line1	Low	1	2	2	Active	
%20and%20(select%20substring(@@version,3,1))%3d'X'	Boiler	Steam_Supplied_To_Line1	Low	1	2	2	Active	
%20and%20(select%20substring(@@version,3,1))%3d'S'	Boiler	Steam_Supplied_To_Line1	Low	1	2	2	Active	
%20and%20(select%20substring(@@version,2,1))%3d'y'	Boiler	Steam_Supplied_To_Line1	Low	1	2	2	Active	
%20and%20(select%20substring(@@version,2,1))%3d'I'	Boiler	Steam_Supplied_To_Line1	Low	1	2	2	Active	
%20and%20(select%20substring(@@version,1,1))%3d'X'	Boiler	Steam_Supplied_To_Line1	Low	1	2	2	Active	
%20and%20(select%20substring(@@version,1,1))%3d'M'	Boiler	Steam_Supplied_To_Line1	Low	1	2	2	Active	

173 total

1 2 3 4 5

© 2021 - 2023 L&T Technology Services Limited. All Rights Reserved.


ESM
Bing
https://www.bing.com/?toWww=1&redig=904820014ECA49E88CC8760B0D4EA043
Microsoft Bing
Chat
Work
Images
Videos
Maps
Search the web
Protecting earth's biodiversity for everyone
PM Modi in Papua New Gu...
Shops cannot decline Rs 2,...
G20 Summit In J&K From T...
'Office of President reduce...
Modi documentary row
Microsoft Search

2.7 Lack of rate limit leads to Brute-force Attack			
Impact	Medium	**Risk Rating**	Medium
Ease of Exploit	Moderate		
Likelihood	Medium		
Category	A6: Security Misconfiguration		
URL/Impacted system	http://13.71.66.245/#/login		

Description

Rate limiting is a technique used in computer systems and network infrastructure to control and limit the number of requests or actions that can be performed within a certain time period. Here we are able to give more than six thousand requests for credentials and the application is not blocking that.

Impact

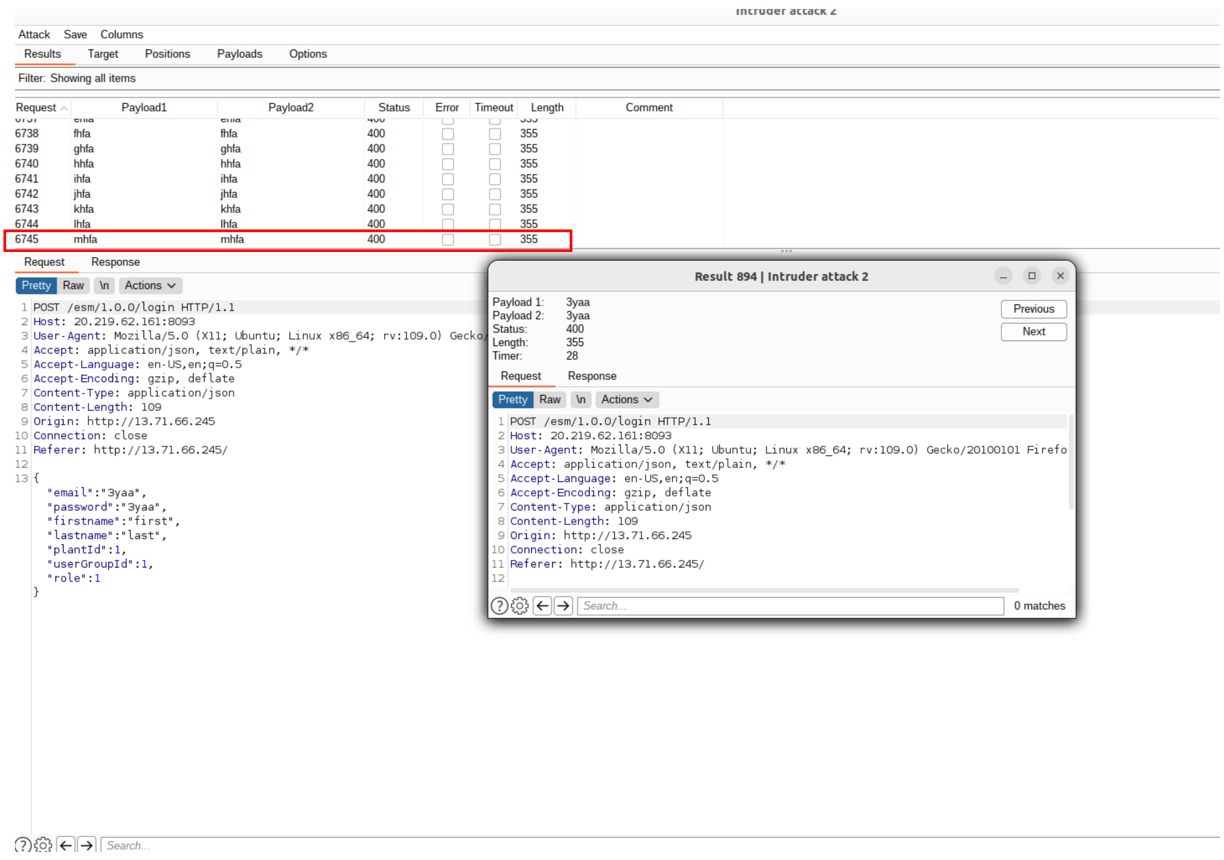
It sees a broad range of applications, from preventing DoS attacks at the proxy level to locking accounts to prevent Brute-force attacks. While it can be admittedly annoying at times, an application without any form of rate limiting is begging to be targeted as there is no limit set to control the requests that can be sent.

Recommendation

To mitigate this issue developers should implement a timeout after several requests in a period or implement a CAPTCHA mechanism on the form page.

How to recreate the Security defect

- Tried to login in as a user account.
- Capture the traffic into the burp suite and send it to the intruder tab.
- Set the payloads (dictionary files [containing username and passwords])
- Click on start attack.

Evidence


Intruder attack 2

Request	Payload1	Payload2	Status	Error	Timeout	Length	Comment
6738	fhfa	fhfa	400	☐	☐	355	
6739	ghfa	ghfa	400	☐	☐	355	
6740	hhfa	hhfa	400	☐	☐	355	
6741	ihfa	ihfa	400	☐	☐	355	
6742	jhfa	jhfa	400	☐	☐	355	
6743	khfa	khfa	400	☐	☐	355	
6744	lhfa	lhfa	400	☐	☐	355	
6745	mhfa	mhfa	400	☐	☐	355	

Filter: Showing all items

```
POST /esm/1.0.0/login HTTP/1.1
Host: 20.219.62.161:8093
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 109
Origin: http://13.71.66.245
Connection: close
Referer: http://13.71.66.245/

{
  "email":"3yaa",
  "password":"3yaa",
  "firstname":"first",
  "lastname":"last",
  "plantId":1,
  "userGroupId":1,
  "role":1
}
```

Result 894 | Intruder attack 2

Payload 1:	3yaa
Payload 2:	3yaa
Status:	400
Length:	355
Timer:	28

```
POST /esm/1.0.0/login HTTP/1.1
Host: 20.219.62.161:8093
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefo
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 109
Origin: http://13.71.66.245
Connection: close
Referer: http://13.71.66.245/

```

0 matches

2.8 No Content Security Policy

Impact	Low	**Risk Rating**	Low
Ease of Exploit	Difficult		
Likelihood	Medium		
Category	CWE-1021: Improper Restriction of Rendered UI Layers or Frames		
URL/Impacted system	http://13.71.66.245/#/login		

Description

During the HTTP traffic analysis of the ESM Web interface, it was observed that the server does not support Content Security Policy. Content Security Policy is a standard that helps protect against various content injection attacks like cross-site scripting. While the victim is interacting with seemingly harmless web pages.

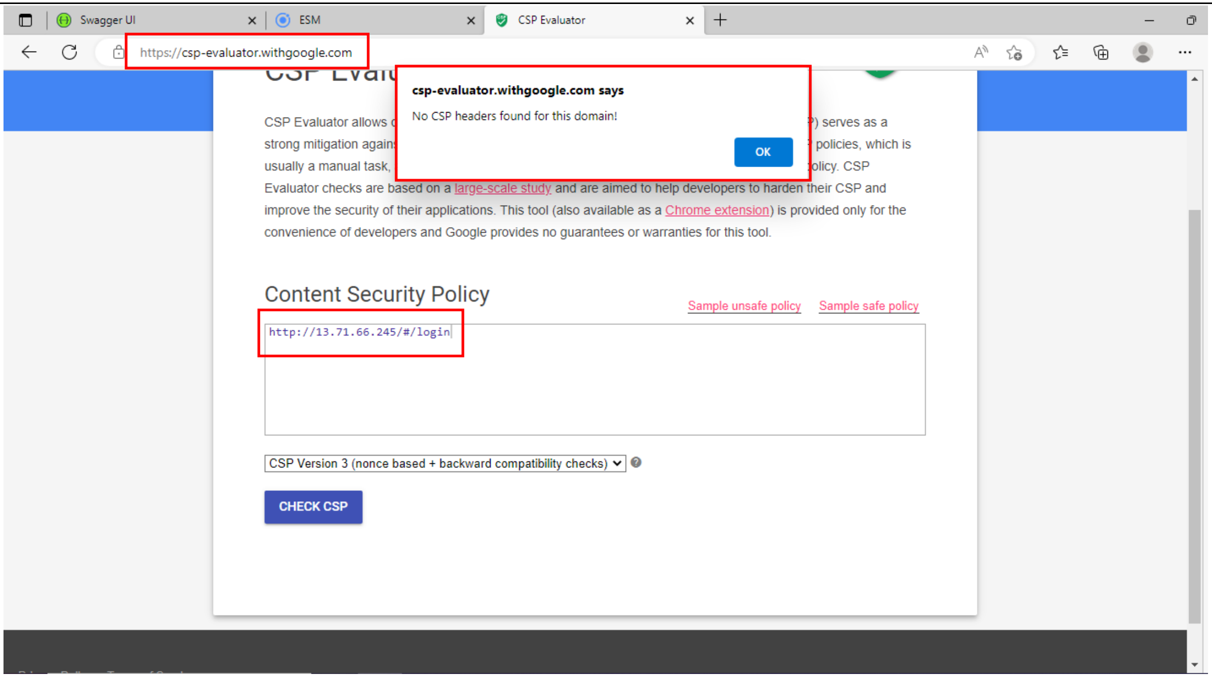
Impact

Without a Content Security Policy, an attacker can perform content injection attacks if data from the service is displayed in a browser.

Recommendation

Enabling the Content Security Policy response header to all HTTP server responses helps in

L&T Technology Services

preventing content injection attacks. While adding Content Security Policy it must be set correctly specifying the locations from which content can be loaded. Content-Security-Policy: <Policy-directive.
How to recreate the Security defect
• Browse to - https://csp-evaluator.withgoogle.com/ • Enter the URL – http://13.71.66.245/#/login • Click on check CSP.
Evidence
 Swagger UI · ESM · CSP Evaluator https://csp-evaluator.withgoogle.com **csp-evaluator.withgoogle.com says** No CSP headers found for this domain! OK CSP Evaluator allows ...) serves as a strong mitigation again... policies, which is usually a manual task, ... olicy. CSP Evaluator checks are based on a large-scale study and are aimed to help developers to harden their CSP and improve the security of their applications. This tool (also available as a Chrome extension) is provided only for the convenience of developers and Google provides no guarantees or warranties for this tool. Content Security Policy Sample unsafe policy · Sample safe policy `http://13.71.66.245/#/login` CSP Version 3 (nonce based + backward compatibility checks) CHECK CSP

2.9 Application Allows Concurrent Sessions			
Impact	Low	**Risk Rating**	Low
Ease of Exploit	Difficult		
Likelihood	Low		
Category	CWE-1018: Manage User Sessions		
URL/Impacted system	http://13.71.66.245/#/login		
Description			
The application allows concurrent sessions, multiple users can login to the application simultaneously with the same user credentials.			
Impact			
Attacker can make victim's account active with the same username and password.			
Recommendation			
The application should only allow a user to establish a single session with a particular set of credentials at a time. Once that session has been established, subsequent attempts to login using			

L&T Technology Services

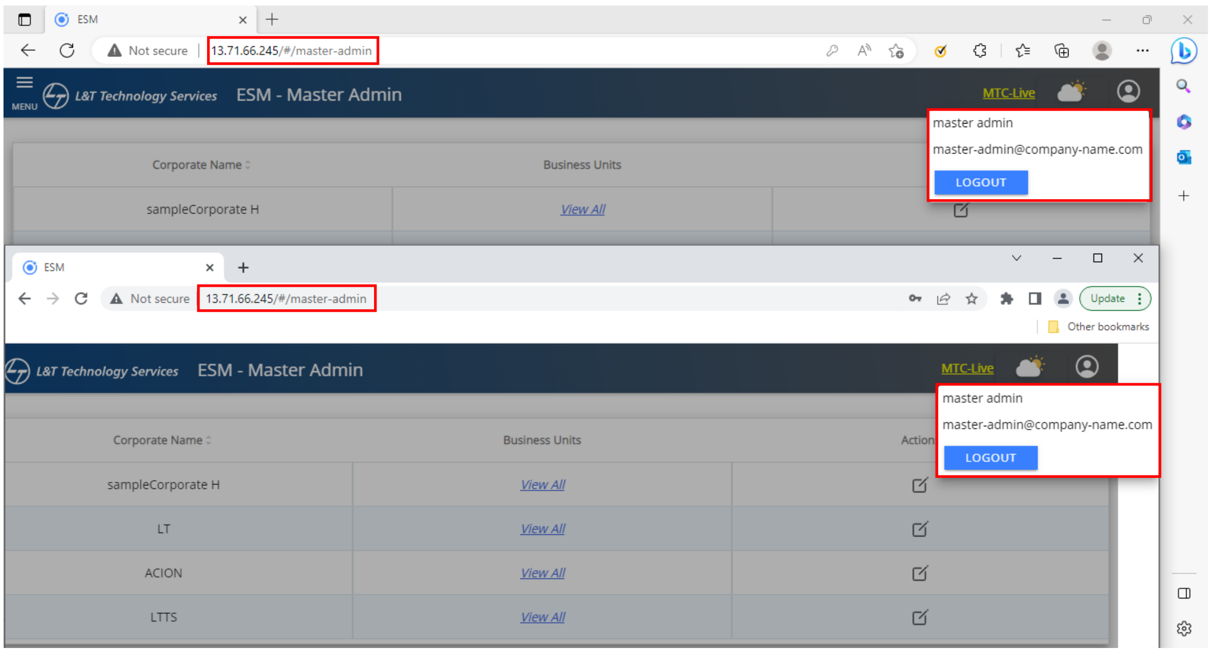
those credentials should either be denied or existing sessions should be terminated, depending on business needs.

If concurrent sessions are required for business purposes, additional session management features must be provided to ensure that all end users are made aware of multiple sessions. Such features include allowing end users to view all current sessions, prompting users when a new session is created, and providing users the ability to terminate unwanted sessions. Additionally, when allowing concurrent sessions, it is recommended that users are notified that their credentials were used to establish a new session, including the time and IP from which the session was established

How to recreate the Security defect

- Login into the application.
- Capture traffic observe response.
- Again, Login in the new browser with same credential as before.
- Notice that we multiple users can login to the application simultaneously with the same user credentials.

Evidence



Corporate Name	Business Units	Action
sampleCorporate H	View All	
LT	View All	
ACION	View All	
LTTS	View All	

2.10 No Multifactor Authentication (MFA)

Impact	Low	**Risk Rating**	Info
Ease of Exploit	Easy		
Likelihood	Low		
Category	CWE-308: Use of Single-factor Authentication		
URL/Impacted system	http://13.71.66.245/#/login		

Description
The application uses single-factor authentication to authenticate privileged users to the system. Single-factor authentication refers to the use of a single component to identify an end user of an application or system. The factor provided may be something the user knows, something the user is, or something the user has. Each of these options provides its own set of advantages and risks when used for authentication: • "Something you know", such as a user-defined password, may be easily created and changed when necessary. Authentication factors derived from the end user must have some degree to be managed by the user themselves, leaving the known secret's security up to them. This can result in the secret being forgotten or exposed through a breach of a separate system that holds or uses the same known secret. • "Something you are", such as a fingerprint, provides an end user with a constant factor that cannot be easily acquired or mimicked by an attacker. While this initially provides a strong barrier to entry and will always be with the end user, a single breach could leave the attribute used for authentication useless as it cannot be updated. • "Something you have", such as a hardware token, can be managed from a central source and is configured to constantly update, removing responsibility for the known secret from the user. However, this transition of the knowledge base may hinder the application's accessibility if the device is not always at hand.
Impact
If an attacker compromises the authentication mechanism (e.g., a victim's account password), they will have full access to functionality and data normally only available to the victim.
Recommendation
Multi-factor authentication should be implemented and enforced for externally accessible applications containing sensitive data or functionality. Multi-factor authentication is built upon the combination of two or more components that can prove a user's identity to the application. This provides an additional layer of security as it is assumed that an unauthorized attacker will not be able to supply both factors required for authentication. The factors required by the application should be a combination of at least two distinct factors from the following: • Something the user knows • Something the user is • Something the user has For example, a common multi-factor authentication mechanism requires a user to provide a password they have created (something they know), as well as a value from a hardware token (something they have). If an attacker can compromise a user's password, they will still not have access to the hardware token and will not be able to gain access to the system. Note: Requiring two or more pieces of information for authentication that fall under the same factor category does not provide true multi-factor authentication. For example, a user's password and the answer to their security question are both something the user knows. Requiring both during authentication does not represent true multi-factor authentication.
How to recreate the Security defect
• Browse application. • Perform any critical operation. • Observe there is no additional authentication step for any critical operations.

2.11 Long Session Timeout			
Impact	Low	**Risk Rating**	Info
Ease of Exploit	Easy		
Likelihood	Low		
Category	CWE-1018: Session Management		
URL/Impacted system	http://13.71.66.245/#/login		

Description

During the assessment, we observed an attacker's ability to hijack a victim's session increases proportionally with the amount of time an idle user's session remains valid. Once a valid session identifier is obtained, the attacker can impersonate the victim in the application, performing any functionality and accessing any data made available to the victim.

Impact

• Insufficient Session timeout increases a web site exposure to attacks that steal or reuse user's session identifiers.
• Cookie Hijacking is possible, application integrity can be compromised.
• Once a valid session identifier is obtained, the attacker can impersonate the victim in the application.

Recommendation

Terminate the user's session server-side after a sufficiently short idle period. When the user makes further requests using the expired session, they should be redirected to a splash page or the login pages. In addition, the client-side code should track session idle time and automatically redirect the user to a splash page or the login page after a certain period of client-side inactivity has passed. No prior authenticated user data or functionality should continue to be displayed after the timeout occurs.

Determine a session timeout duration that sufficiently protects end users and the application while maintaining system usability. Session timeouts of 30-120 minutes are common for most web applications and vary depending on the sensitivity of the information available during each session. Various standards organizations and government entities also typically recommend organization-defined timeouts or call for an idle timeout of 30-120 minutes:

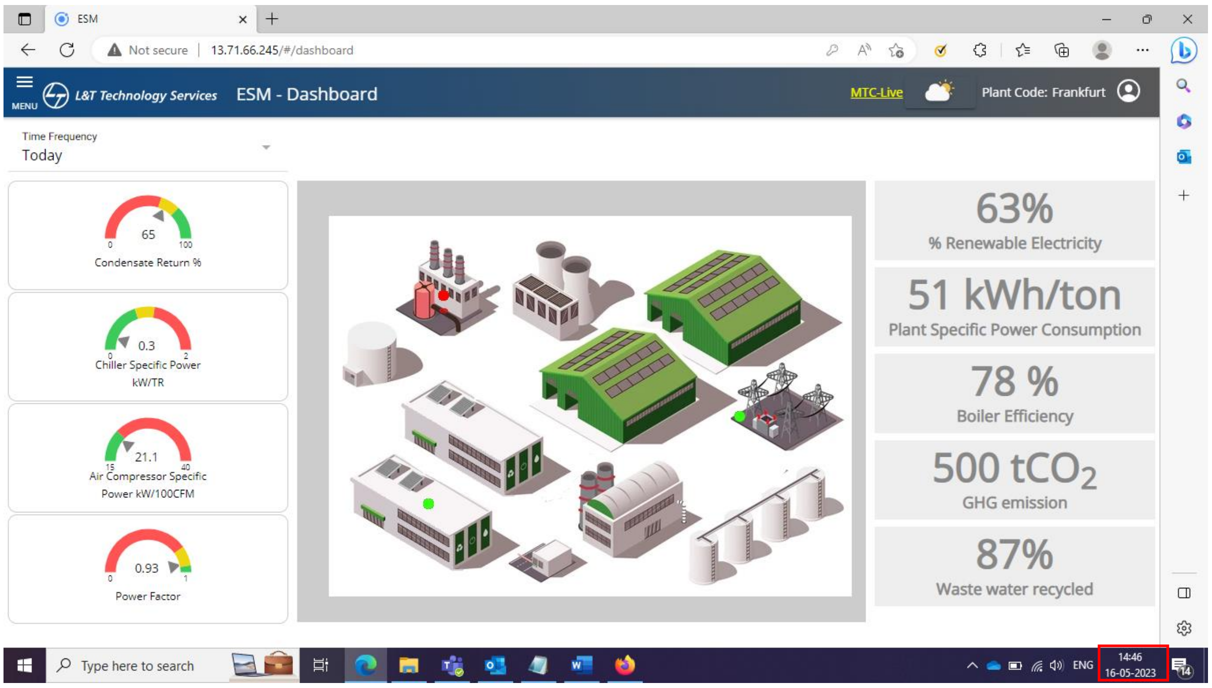
- PCI DSS v3.2 section 8.1.8 states, "If a session has been idle for more than 15 minutes, require the user to re-authenticate to re-activate the terminal or session."
- U.S. CNSS - CNSSI No. 1253 section AC-11 states, "Session Lock ... not to exceed 30 minutes"
- NIST SP800-53 section AC-11 states, "...Prevents further access to the system by initiating a session lock after [Assignment: the organization-defined period of inactivity or upon receiving a request from a user".

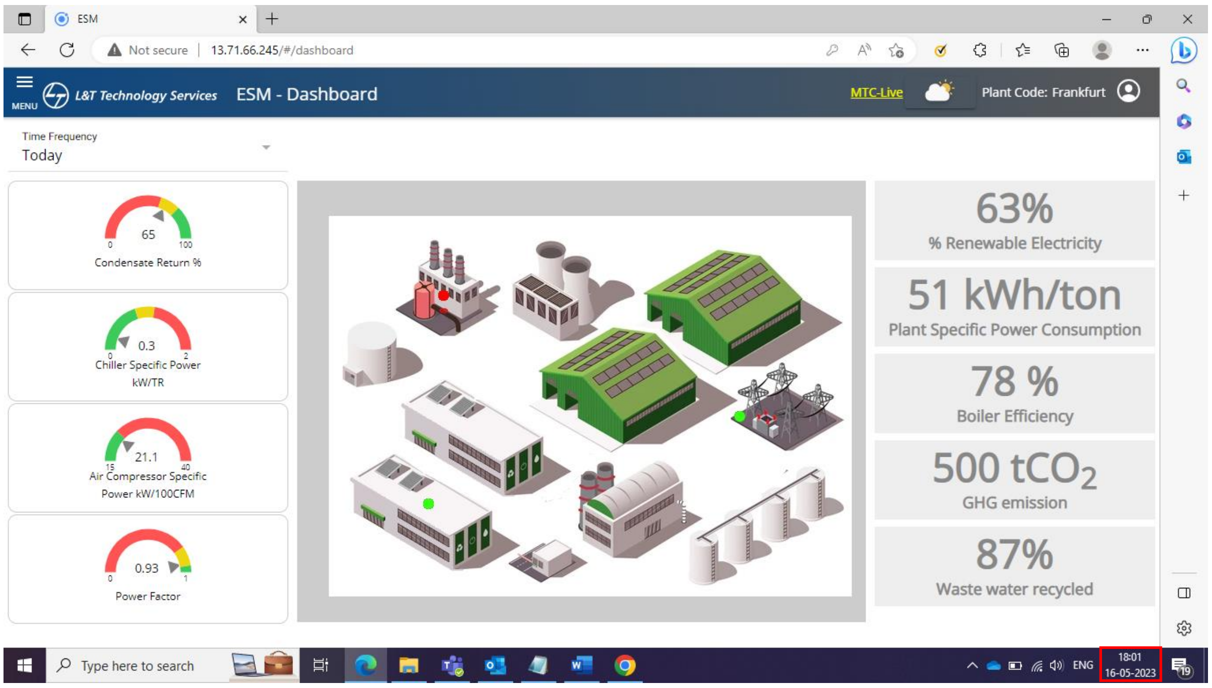
How to recreate the Security defect

• Login to the application.
• Idle the application for up to 30-60 minutes.
• Can access the application even after 30-60 minutes.

L&T Technology Services

Evidence





L&T Technology Services

3. Abbreviation

APP	Application
HTML	Hyper Text Mark-up Language
HTTP(S)	Hypertext transfer protocol (Secured)
Pg.	Page
TLS	Transport Layer Security
SSL	Secure Sockets Layer
IP	Internet Protocol
LTTS	Larsen & Toubro Technology Services
OWASP	Open Web Application Security Project
VAPT	Vulnerability Assessment and Penetration testing
CSP	Content Security Policy
IDOR	Insecure direct object references
MFA	No Multifactor Authentication
URL	Uniform Resource locator
XSS	Cross-Site Scripting
XXE	XML External Entities
SQL	Structured Query Language